UNIVERSITY OF MINNESOTA COMPUTER CENTER
Deadstart Systems Newsletter

24 June 1980 Vol. 6, No. 12

Send all comments, criticisms and contributions to the editor: T. W. Lanzatella
University Computer Center, 2520 Broadway Drive, Lauderdale, MN 55113.
The University of Minnesota is an equal opportunity educator and employer.

TABLE OF CONTENTS

NOTICE OF CHANGES TO THE SYSTEM 102

PROPOSED CHANGES TO THE SYSTEM 103

UC OPTION FOR PURGALL - K. Fjelsted 103

SYSTEM MAINTENANCE 104

LAST WEEK'S SYSTEMS GROUP MEETING - T. W. Lanzatella 104
CALLPRG AND LIBRARY TAPE NEWS - M. Riviere 106
LETTERS TO THE EDITOR ON POLISHING OUR SYSTEM - T. W. Lanzatella 106
CYBER 720 DEADSTART DUMP ANALYSIS - R. A. Williams 107
REVERSE POLISH (PART II) - D. W. Mears 108
CYBER DEADSTART DUMP ANALYSIS - K. C. Matthews 117

NOTICE OF CHANGES TO THE SYSTEM

Tom Lanzatella changed the ACCOUNT CLOSED dayfile message to refer to Jack Schwab instead of Judy Eakman. The modset MAINT and common deck COMPCPA were removed thus removing the MAINTENANCE subsystem.

Kevin Matthews repaired two errors stemming from the recently installed DSP.

a) Output files with a certain file ID were always printing at Lauderdale regardless of their origin. This ID happened to be used by ACCSTAT hence, jobs using ACCSTAT printed at Lauderdale.
b) Internal calls to deferred ROUTE a file to an illegal terminal ID resulted in an incorrect error code in the response. This confused PLOT31.

Don Mears removed the IQFT mechanism from DELAY queue processing. Although this change was not supposed to occur until the end of summer, recent problems with DELAY queue jobs not running made the change necessary. On two occasions recently the DELAY queue was not brought in during LOWRATE or NOFRILLS hours. Don suspects that this may have been caused by track limits on the shared queue device. This is an extremely complex problem and time was a critical factor. Since the option to remove the mechanism was approved we took the option. The only effect this change has on users in that E,JN will now indicate a users DELAY jobs. The QUEUE utility still returns the same information as before.

Bill Sackett changed the DTKM monitor function to hang if the track to be dropped belongs to a preserved file and the track is not the first

track of the chain. Bill also changed SYSEDIT to be a little smarter about placing things in ECS when ECS has very few tracks.

Jeff Drummond added an assembly option to COMPCHI which allows channel instructions in several QUAL blocks to assemble correctly. Jeff also repaired a few errors in TRANSIT related to protected permanent file detection and FET relocation.

Brad Blasing changed 1AJ/LDR to disallow special entry points (RFL=,SSM=,etc.) as starting points for overlay loads. This had been a breach of execute-only file security.

Dean Nelson corrected DSD and USERS/DSD to properly reference a table of job classes. Some staff may have noticed that the J-display was putting out garbage for certain classes.

Arnie Nelson added two new sites to SUPIO.

PROPOSED CHANGES TO THE SYSTEM

UC Option for the PURGALL Control Statement - by K. Fjelsted

With the advent of the UC option on the permanent file commands save, retain, define, change, and catlist, users have the ability to catagorize their files by assigning a single user control word (one to seven characters the first being alpha, and the remaining being alpha or numeric). This has helped those of us who work on several projects at the same time, or for reasons of organization wish to group our files.

There have been many times when upon the completion of a project I have wanted to rid my permanent file space of that group of programs, test data, procedure files, and the like. If I desire to do this I must explicitly purge each one of these files.

I propose to add to the PURGALL control statement the capability of handling this kind of purge operation. The purgall statement already possesses the ability to purge all files which can be clasified as direct access, indirect access, created, modified, or accessed before a specific date or time, private, semi private, or public, or the entire catalog. I propose to add a UC option to the purgall statement which when specified would purge all files which match the user control word given. In order to purge only the files which have their user control word set to ABC one would execute the control statement, PURGALL(UC=ABC). One will still be able to mix in the other options which are available on the PURGALL statement. This would make it possible (as an example) to purge only the direct access files which have the user control word ABC but leave the indirect access files alone.

SYSTEM MAINTENANCE: People and Procedures

Last Weeks Systems Group Meeting - by T. W. Lanzatella

The following proposals were discussed.

1. Tom Lanzatella's proposal to eliminate the use of IQFT for storing Delay jobs was approved (see DSN 6,11 p. 85). We stipulated that the change should not occur until the end of summer due to the number of things which have to change like writeup DELAY and the QUEUE utility.

2. Tom Lanzatella's proposal to include connect-time among the COST dayfile message was rejected (see DSN 6,11 p. 85). This type of information was considered inappropriate for COST since connect-time is a supplies item where all other costs reported by COST are not.

3. Tom Lanzatella's proposal to change SEND so that a DA file will be automatically copied to a local file was approved (see DSN 6,11 p. 86).

4. Kevin Matthews' proposed scheme for keeping track of deadstart dumps taken during system time was approved (see DSN 6,11 p. 86).

5. Dean Nelson's proposal to remove the PACKMS utility was approved with the following stipulations (see DSN 6,11 p. 86):
   a. WRITEUP(CONTROL=PACKMS) will be removed.
   b. WRITEUP(PACKMS) will be changed to indicate that PACKMS is no longer available and will give several examples of how users can achieve the same results using the standard random file subroutines in Fortran.
   c. After one year, WRITEUP(PACKMS) will be removed and if usage warrents, examples will be added to WRITEUP(FORSUBS=WRITMS).
   d. A UCC newsletter article will be written informing users of the removal of PACKMS and of the examples offered on WRITEUP(PACKMS).

6. Andy Hastings' proposal to add master user access to subordinate user's tapes was approved (see DSN 6,11 p. 87).

7. Tim Hoffmann's proposal to reorganize the way that maintenance jobs are automatically scheduled was discussed and rejected (see DSN 6,11 p. 87). We decided instead to scotch the entire modset and go back to running a job submitted by ISF.

8. Paul Thompson's proposal to add INQ,DEL and MI options to RELOAD was approved but we thought that INQ should be ENQ since inquire seems to be spelled with an e throughtout the system (see DSN 6,11 p. 87).

9. Bill Sackett's proposal to have LDR issue statistical account file messages whenever secondary overlays are loaded from the system was approved (see DSN 6,11 p. 87). We decided to use a message prefix different from ZLSY and to key off a new PMS bit.

10. Bill Sackett's proposal to change DTKM to hang if a preserved file track is being dropped and the track is not the first track of the chain was approved (see DSN 6,11 p. 87).

11. Mike Frisch's proposal to truncate the file DAYFILE when appending a dayfile was approved (see DSN 6,11 p. 88). Jeff Drummond will study the implementation. We spent a considerable amount of time haggling over the message that would be added to the end of DAYFILE. The following text was eventually wrung out: FILE DAYFILE IS TOO LARGE AND HAS BEEN TRUNCATED. SEE WRITEUP(SUBMIT) OR (SEND).

12. Mike Frisch's proposal to install a control statement preprocessor which would check for syntactic errors was approved (see DSN 6,11 p. 88). We decided however that the project was inappropriate for the systems group and that Mike can do the project if he can find the manpower.

13. The John Larsen/A. Swanson proposal to add a ND (no drop) parameter to the Callprg index entry was narrowly rejected (see DSN 6,11 p. 88). We all recognized the utility of the option. Marisa felt that the change should be considered after WRITEUP and CALLPRG are separated when serious consideration can be given to CALLPRG usability enhancements.

14. The John Larsen/A. Swanson proposal to add UN=$ and PN=$ options to the Callprg index entry was defeated. We felt that the same facility could be obtained using procedure files.

Larry Liddiard spoke briefly about software pricing practices.

a) We have to think in terms of fair exchange when we send out locally developed software. A price should be established for all locally written packages.
b) Larry is soliciting opinions on what a consistent policy might be.

Larry Liddiard described the changes he sees coming for summer 1981.

a) We need immediate relief from saturation - memory on the C74, PPU's on the 172.
b) Hoping for a new computer in 1983-1984.
c) We must provide compatibility for existing software.
d) We must cut our operating costs.
e) A good possibility that the C74 is replaced by a C750 or C760 in summer 81.
f) We need to provide VAX service by summer 81.
g) We need wide band support between Lauderdale/East Bank/West Bank by summer 81.
h) Time sharing service is expanding at the rate of 32 posts per year. We will require 240 ports by 1984.
i) Maybe an IBM machine in 1982.

//////////

CALLPRG and Library Tape Changes - by M. Riviere

On July 1, S. Yen will be updating his Callprg index package SPSSONL, by making the future version current and the current version past. The change will be taking place on the three Cybers.

The next set of Callprg and Library Tape modifications will be taking place on July 10th. Modifications for that date should be submitted before noon, July 3.

//////////

Letters to the Editor on Polishing Our System

Here are two more responses to the Polishing Our System article.

From Phil Kachelmyer:

Andy Mickel's article in the 20 May 1980 DSN was interesting and thought-provoking. I also would like to see some of the things he spoke of done. It is distressing to all concerned to have a user come up to me at the consultants desk with a USER card that simply didn't have a period at the end and not know what ILLEGAL CONTROL CARD means. Other users are generally kept waiting in line. I am irritated by having to answer the same question for the 83rd time today. And the user is embarrassed at the silly error. I feel that most of Andy's other comments perhaps should be considered as well. I realize that there is probably much more work involved than meets the eye, but by implementing some of these changes, perhaps the sanity of a few users and consultants would remain intact.

//////////

From Joe Jaynes:

Yet another response to Andy Mickel's "user friendly systems" article.

I wholeheartedly support the struggle to make our operating system more coherent, logical, and easy to use. While much of the difficulty no doubt lies squarely in CDC's camp, I suspect that great strides have been, and must continue to be, made locally. To me, as to the others who have written responses, the two most critical areas remain control statement clarity and comprehensible error messages. Particularly the latter: there is absolutely no reason for users to be mislead when an error message, by definition, is supposed to illuminate rather than obscure the problem.

But, I feel that the whole question of "polishing the system" is perhaps misdirected, considering the rapid turnover in hardware and attendent operating systems. It seems that the question is really one of design priorities. Are we at UCC providing the best possible services -- both in terms of hardware and other, less tangible products -- to our users? In certain areas we are; but in others we fall short. For users familiar with the intricacies of NOS, or those who frequently do large, repetitive

number-crunching runs, the operating system and changes to it are of minor concern. But there are other users, of whom students form the largest portion, who suffer completely unnecessary inconvenience (to say nothing of emotional trauma!) at the hands of the current system. And those agonies are compounded every time we upgrade! For them, improvements in CPU speed, and CM and peripheral storage capacity hardly compensate for the new confusions in their control card decks and dayfiles. In other words, it is not a question of our system being "user friendly" so much as "friendly to whom?"

Consequently, I see this issue as one of deep-seated priorities rather than of surface capabilities. In the maintenance of our current system, and in the selection of new ones, we must constantly be aware of the needs of all our users, be they government agencies or students doing homework programs. We must try, as much as possible, to accomodate all of their needs. I realize that everbody can't be happy, but we can do a better job than we are doing now. To do less is to shirk our primary responsibility: to provide superlative service to our user community.

//////////

Cyber 170-720 Deadstart Dump Analysis (6/9-6/19) - by R. A. Williams

Date	Description	Tape
800611	The system went down when the CDC engineers, who were preparing for installation of the ECS coupler, made some changes in the mainframe.	N.A.
800618	A freon leak in the mainframe followed by a disk drive breakdown caused about 10 hours of down time.	Fixed
800619	The scopes went blank 5 different times during the day when the CPU hung up. The problem was diagnosed and fixed by the CDC engineers. The system was trying to execute an ECS read instruction for new style ECS. Since that equipment doesn't exist on the machine, the CPU was hanging up on it.	Fixed

//////////

Reverse Polish (Part 2) - by D. W. Mears

Part 1 of this article discussed in general terms why we should not attempt to do the "polishing" Andy suggested in his "Polishing the System" article in DSN 6,11 pp. 74-82. It gave counterarguments to the first few suggestions in order to show that the "improvements" were subjective and debatable.

This article gives some counterarguments for the remainder of the "polishing" suggestions. It is important to respond to these suggestions for several reasons: The submittors deserve a written response in return for their work of collecting and writing the suggestions. The implied or inferred idea that all or most of the suggestions are worthwhile, would improve the system, and should be installed must be disspelled. By pointing out what is wrong with most of these suggestions some people may be better able to make useful suggestions in the future.

In addition to the specific problems each suggestion has, there are general classes of problems common to many of the suggestions. These problems are apparent not on first reading, but only when the suggestions are analyzed. The most troublesome area (which is common to almost all of the suggestions) is that the suggestions are worded as solutions to obvious, but unstated, problems. In many cases, I could not figure out what problem was supposed to be solved by the suggested solutions. If people want the system group to solve a system usability problem, they should expend more words and effort detailing the problem.

There is serious lack of logic behind the suggestions. This is especially true for the suggestions to change messages. The suggestions go: change XXX message to YYY because YYY is clearer and better. The trouble here is that "clearer" and "better" are subjective terms. The systems group has trouble agreeing on facts and numbers; arguing for clearer and better is hopeless. The arguments must be made with logic. For example: Change XXX to YYY because XXX is wrong for the following reasons... and YYY is correct, or XXX is confusing because it can be read several ways and YYY is accurate because it can only be read one way.

Many of the suggestions are simply not well researched or thought out. People are suggesting changing things without bothering to find out why things work as they do. They argue that more commands and parameters should work as intuition suggests. The problem is that everyone's intuition is different, so that it is better to have commands and parameters be logical and consistent than intuitively obvious.

There are some suggestions which have nothing really wrong with them except that they request new features which take a significant amount of modification to the operating system to implement correctly. Many of Dick Rubenstein's suggestions fall into this catagory. While there is nothing wrong with this type of suggestion, it is important to ask: Are the benefits of this new feature really worth the effort involved to install the feature?

The following list of counterarguments should not be viewed as reasons why DWM does not like the suggestions (in fact, I agree with few of the suggestions I argue against), but as reasons why people in favor of the suggestions must think and argue more carefully and logically before they suggest or agree to "polishing" changes.

Make DC=PR on route default (instead of DC=IN or DC=SC).
The ROUTE DC parameter currently has three default values. If the specified file was deferred routed, the default is the previously specified DC. If the file was not deferred routed, but is one of the special file names, OUTPUT, PUNCH, PUNCHB, OR P8 (names which carry implicit deferred routing), the default is the DC implicitly associated with that file name (i.e., PR, PH, PB, or P8, respectively). Only if the file is not explicitly or implicitly deferred routed is the default DC=SC. Thus, the absence of a DC parameter means route the file using its explicit or implicit deferred routing and (as a logical conclusion) if no deferred routing is set, do nothing with file. This seems to me to be a logical and consistent way to handle the default DC. Changing any combination of the default values of DC to DC=PR will eliminate this consistency and will cause confusion. If only that case which now defaults to DC=SC is changed to default to DC=PR, users who get used to omitting DC will get burned by the implicit routing of PUNCH, PUNCHB and P8 named files (their print file named P8 will end up on the punch). If the DC=SC default and the punch default for PUNCH, PUNCHB and P8 are changed to DC=PR, users who believe the NOS RM and make the reasonable assumption that "ROUTE, PUNCH." will send their punch file to the punch will be surprised when it ends up on the printer. If all three defaults of DC are changed to DC=PR, we lose the capability to route a file to wherever it was previously deferred routed. Currently, when PLOT31 is run from a batch job, a user can do "ROUTE,PLOTS." to send the PLOTS file (which was deferred routed by PLOT31) to the plotter. If the defaults are changed, this ROUTE will send the PLOTS file to the printer. To summarize, the current defaults are logical and consistant. Changing the defaults (to save six key strokes) will necessitate additional UCC documentation which contradicts the NOS Reference Manual, will cause some confused users to send files to the wrong devices, and/or will remove existing capabilities.

"DELETED" is an incongruous response to escape key...
Here are three solutions to this incongruity:

1) Change the message from *DEL* to something else. *DEL* is a reasonable message from a functional standpoint. The asterisks tend to make the message stand out from the previously entered text so that when timesharing session listing is reviewed, it is clear when an input line was deleted. "DEL" is a good short mnemonic message to indicate that the entered line was deleted. The message must be short in order for the escape key to be useful. The escape key is used to save the time it would take to backspace the appropriate number of characters. The longer the message is, the less time is saved and thus, the less useful the escape key becomes. This is especially true on 10 cps terminals. For this reason, the suggested message "LINE DELETED." is functionally inferior to *DEL*. *ECS* has also been suggested as a replacement for *DEL*. The problem is that the BREAK key also

deletes input lines. It would be more incongruous to have BREAK generate *ESC* than to have the escape key generate *DEL*.

2) Move the line delete function from the escape key to some other key (e.g. the DELETE key). This is bad because the escape key is so well positioned and easily accessed on most terminals (usually protruding from the left side or in the corner of the keyboard). Whereas the DELETE key, for example, is usually surrounded by other keys on the keyboard. For this reason many users would object to moving the line delete function to another key.

3) Modify all terminals which have a key labeled DELETE. UCC could prepare pressure sensitive stickers labeled RUBOUT and place them over the DELETE keys. This is certainly the most general solution and it has the advantage of not impacting functionality or requiring any significant re-education of the users.

I can not believe there is enough confusion over this "incongruity" for us to make any changes in this area.

No way to provide special-request info for submit jobs.
Operations is trying to eliminate the need for special request info.

Fix problem of 2 translations for ^ and @.
We must not create a new character set. A new character (even if only a few characters were translated differently) would only add to the present character processing problem. Our basic problem is that we are attempting to represent seven bits of character information in 6 bits. No matter how much we modify our character set, this problem will remain. A character set change is a traumatic experience for users who must modify their programs and convert all their files and it is a lot of work for UCC staff who must do the system modification and provide the conversion documentation. Solving the problem of two translations for @ and ^ certainly cannot justify a conversion effort. We would be much better off directing this effort towards providing full 8-bit ASCII support (even if it is wastefully implemented and reduces efficiency). Any loss of efficiency should be viewed as the cost of doing things right.

Change terminology "normal/extended" to something like "subset-ASCII/full-ASCII."
There should only be one set of terms describing the character set mode even if that set of terms is not entirely accurate. It is better to have one bad terminology than two terminologies - one better than the other. The problem with the character set terminology is that we are attempting to describe two pairs of attributes with only two words. One set of attributes describes the binary codes being used. These are the normal 6-bit display codes and the extended 6/12-bit codes to handle the extension to 63 character set. The other set of attributes describes the actual character graphics being represented. These are subset-ASCII for the 63 ASCII characters which are being represented by normal display code and full-ASCII (a term which usually implies 8-bit ASCII) for the full 128 ASCII characters which are being represented by the 6/12-bit extension to display code. Thus, CDC describes the character set mode by using the binary code attribute "normal" for the 63 character set and

using the character graphic attribute "ASCII" for the 128 character set. This may be confusing, but it is not clear that always using the character graphic attributes subset-ASCII and full-ASCII is much better.

RETURN, COPY, etc., should warn when a file is empty.
A more general solution is to have all system routines indicate how much data was processed. This would provide positive feedback in all instances - not just in the case where the file is empty. There is no reason for RETURN to issue a "file empty" warning message. Since RETURN does not process data there is no logical reason for it to even check for a file empty condition. A more serious error than file empty is the attempt to read a file which is not assigned to the job. It is this case (not file empty) which usually indicates that the user has done something wrong (misspelled a file name or forgotten a GET, for example). Logically this should always produce a "FILE NOT FOUND" message.

Eliminate extraneous and oddball control statements.
Andy suggests removing AMEND and EDIT, for example. However, AMEND and EDIT do things XEDIT does not do. AMEND will copy all lines containing a specified string to another file and will perform UNBLOCK type functions. EDIT has extensive elipse string and text alignment processing which is not available in XEDIT. AMEND's usage is probably low enough (about 10 accesses per month) to justify its removal. However, we should give users an adequate warning of the removal and attempt to provide equivalent features in XEDIT. Andy is wrong when he says EDIT is "hardly used at all." The March statistics show that EDIT was used 876 times, equally distributed among all three machines. This is in the upper fourth of all control statements. Removing a product this heavily used would upset many users. We must seriously consider the impact on the users when we make our ease of use "improvements" - especially when our "improvements" remove existing capabilities.

The NOS-485 change to NORMAL is a step backwards.
The only change to NORMAL at 485 was to make it clear BRIEF mode. Since the function of the NORMAL command is to set all terminal modes back to "normal", this is a logical extension. This certainly does not seem like a step backwards. Andy's argument that "NORMAL is useless because of all its side effects" indicates that he simply does not understand the function of the NORMAL command. It is true that TAPE mode cannot be cleared without also clearing CSET, ASCII mode, but this has always been true in NOS and KRONOS. It did not change in NOS-485.

The BRIEF command should toggle brief mode.
There are no timesharing commands or control cards which toggle modes. It is logical that typing BRIEF causes BRIEF mode to be entered. It is not logical to have typing BRIEF cause BRIEF mode to be exited. Although the NORMAL command is not an ideal way to clear BRIEF mode, it is acceptable since it is unlikely anyone would want to exit BRIEF mode.

PRINT should not have been replaced by NOTE which rewinds OUTPUT.
The most common use of PRINT I am aware of is to issue informative messages from a procedure file to a timesharing terminal. NOTE accomplishes the same thing with little additional effort. For example, when running from timesharing "PRINT. DONE" is replaced by "NOTE./DONE." The fact that NOTE rewinds OUTPUT does not affect the timesharing user. However, since rewinding output is dangerous for batch jobs, NOTE will probably be changed by CDC or UCC to not rewind OUTPUT by default.

"TYPE *CR* TO CONTINUE" should be changed....
The message could be improved, but I don't understand how Andy came up with the replacement message. The replacement message should not be in lower case because no other operating system program issues lower case messages. This would be an inconsistency and would cause confusion. Andy replaces *CR* with "carriage-return key", but none of the terminals I can find has a key labeled "carriage-return". All of the return keys are labeled "RETURN." When someone wants to change a message they should explain what is wrong with the old message and why the replacement message is better. Andy does not explain what is wrong with the original wording or how the new message is better.

"WRITEUP, anything=*" should work on non-indexed writeups.
There should probably be a way to ask for an entire writeup whether it is indexed or not. "WRITEUP,anything=*" is one way of achieving this, but it would be simpler and more straight forward if "WRITEUP,anything" always listed the entire writeup. The "... NOT IN CORRECT FORMAT" message is not correct and it should be fixed. The message suggests something is wrong with the writeup file - not the control card.

Improve all acknowledging dayfile messages.
This suggestion is too vague and all encompassing to be useful. We need to know which messages are bad and what is wrong with them. We need to know what they should be replaced with and why the replacement is an improvement. Andy's example of changing "EOI ENCOUNTERED." to "END OF INFORMATION ENCOUNTERED." is not an improvement. Anyone who does not know that EOI means end of information probably does not know what an end of information is. Furthermore, "EOI ENCOUNTERED" conveys little if any useful information. The message should be replaced by one which indicates what happened before the EOI was encountered.

Purging a DA file which is attached should change it to a local type file.
This is impossible because more than one user can have the same direct access file attached. When the file is purged, the file type must remain "PM" so that the system knows to decrement the active user count on that file when the file is returned and to release the tracks assigned to the file when the user count goes to zero.

Allow UNROUTE and UNSUBMIT.
Some form of UNROUTE and UNSUBMIT will be available soon (you will at least be able to purge these files).

Process full ASCII punch cards (as CRAY does).
Since CRAY does everything in 8-bit ASCII, it makes sense that they would process full ASCII cards. We will probably never support full ASCII cards because there are no full ASCII keypunches around campus (and probably never will be) and there are so many other areas of the system where full 8-bit ASCII support is more badly needed.

The 40-character dayfile message limit creates a poor user interface.
Dayfile message processing involves an interaction between PPR, MTR, 1DD and mass storage error processors which is hopelessly complex. The 40-character limit is dictated by the lack of space in PPR and the PP communication area. For these reasons, it is not practical for us to attempt to change this limit. Furthermore, there are reasons why the 40-character limit is good. The 40-character limit forces all messages to have the same terse and concise (albeit unfriendly) quality. Any superfluity must be eliminated in order to fit as much accurate information as possible into 40 characters. This serves two purposes. The concise message is easier to read and comprehend because it saves the user the trouble of separating the extraneous words from the "meat" of the message. Secondly, the concise message speeds up the human/machine interaction because it can be printed faster, read faster, and comprehended faster than a similar message with more verbiage. Although the system issues many bad messages, the cause of the bad message is not the 40 character limit. Most bad messages can be replaced by good messages which are less than 40 characters long, and those that cannot probably require a paragraph of text or a writeup rather than an error message to explain the problem. (In NOS-518 CDC completely rewrote dayfile message processing to allow CPU programs to issue 80 character dayfile messages dayfile messages and to allow PPU programs to issue 50 character dayfile messages).

"X," to enter a control statement is a poor user interface.
Both Bill Sackett and Tom Rindflesch misunderstand the reasons why the X command works as it does. The X command is not a kludge and it was not added for reasons of efficiency. In the stock NOS system the X command is valid only in the BATCH subsystem. Its purpose is to allow users to enter control cards which have the same name as TELEX commands. For example, if you wish to execute a local file named BYE, you must enter "X,BYE". Tom suggests that the system should be able to figure out that you wish to execute the control card rather than the TELEX command. But, even if the system could do this, it would not solve the basic problem which is, when a command is valid as both a TELEX command and a control card, the user must be able to execute it either way. (In the example, the user should be able to log off with the BYE TELEX command even though he has a local file named BYE.) The X command is a logical solution to this problem.

In the stock NOS system it is impossible to enter batch control cards from any subsystem other than BATCH. The philosophy is that the language oriented subsystems should contain only the basic de facto standard time-sharing commands most users need for running programs. This limited subset of commands isolates the user from the plethora of batch control statements and makes the system easy to learn and use for the novice user and for the

user who has switched from another timesharing system. If a user enters a command which is not in this subset, TELEX will give an immediate "ILLEGAL COMMAND" response rather than firing up some possibly inappropriate batch-type routine. At UCC we have retained this philosophy, but extended the X command to be legal in all subsystems so that if control cards are needed, they can be entered without the bother of switching to the BATCH subsystem and back. If we remove the requirement of using the X command to enter control cards, we are further lessening isolation of the novice from the more complicated and confusing parts of the system.

The user should be prompted when in doubt or error.
I certainly do not want to be prompted every time I make an error. I usually make many simple, easily corrected errors. The prompting would get in my way and slow me down. It is not clear how the system would determine when the user is confused. How does the system differentiate between a user who is confused and one who is just making typing errors?

Users should never see an octal dump.
The real problem is that users should never need to see an octal dump in order to figure out what has gone wrong. Unfortunately, there are instances where the octal dump provides helpful information. Although the dump may not be useful to the user, it may be useful to the user's consultant to locate the user's problem or a system bug. By making it impossible for the user to get an octal dump, we would be removing this "last resort" method of locating problems.

The message "ILLEGAL TERMINAL" is not very informative.
I think most people understand what this message means in spite of its vagueness. If someone wants this message changed they should explain what is wrong with this message, what a replacement message should be, and why the replacement message is better.

Eliminate the slash from control statement syntax.
The "elimination of slash" argument can be seen as an argument between the novice user and the experienced user and as an argument which typifies much of the "user friendly" debate. The novice user says it makes sense to type GET,A,UN=ABC and that if multiple files are needed, it is better to have to use multiple GET statements than to have to remember a screwy syntax. The experienced user says once you learn the syntax, you can save a lot of time by specifying multiple file names on the control card. We have to ask which is better, a syntax which is more user friendly and easy to use or a syntax which provides more functionality.

Overhaul the control statement language to make parameters consistent and mnemonic.
A couple years ago CDC circulated a Design Action Paper (DAP) which described how they planned to standardize all the control card parameters to make them consistent and mnemonic. Although I do not remember any details of the DAP, I believe that none of the planned changes to existing control card parameters has been implemented. Perhaps, CDC and/or the DAP reviewers felt that although there was a problem and the change was needed, the improvement would cause more confusion that it was worth.

X command should be eliminated.
Minimize differences between direct and indirect files.
Automatically pack primary file on exit from text mode.
OLD and NEW should not clear local files.
I responded to these points earlier.

LIST, LNH, RUN, and RNH should not have equivalenced parameters for non-primary files.
I assume this means these commands should have non-equivalenced parameters for non-primary files as in LIST,lfn or RUN,lfn. If the suggested change is made, it will become easier to RUN and LIST non-primary files, but the syntax for RUN and LIST of the primary file will suffer. For example, RNH,T will change to RNH,,T and LNH,100 will change to LNH,,100. The problem is that RUN and LIST are usually used to run and list the primary file, and so the effect of this change is to make the syntax easier to use for a less frequently used case and harder to use for the more frequently used case. (To accurately determine which case, in fact, occurs more often would require changes to TELEX to record these statistics.)

RETURN should be renamed to RELEASE.
RETURN (which means to give back to a previous owner) does what its definition suggests. It gives files and the resources they represent back to their previous owner, namely the operating system. Release means to free from restraint. I don't think of files as being restrained when they are assigned to a job.

NOS should remember if a file is full ASCII or NORMAL.
I discussed this in my response to the suggestion to eliminate the need for EC=A9 on ROUTE.

File names should be more than 7 characters.
It is almost impossible to change the number of characters in local file names because of the large number of programs which depend on seven character names. It might be possible to increase the number of characters in permanent file names when we increase the size of the catalog entry to 16 words, but it would take a lot of work.

To make this change we would have to change the definition of the fields in the FET to accomodate the larger permanent file name, rewrite PFILES and CATLIST parameter processing to allow longer names on the control card, reformat the report produced by CATLIST and CATLSYS, and make some major changes to PFM. If we ever did get it to work, most user programs which reference permanent files would stop working because everyone assumes the entire name will fit in one 60 bit word. The increased convenience of having longer file names can not justify the amount of work the change would require and the amount of trouble the change would cause.

A hierarchy of file organization is needed.
Permanent files can be grouped for some purposes through the use of the "UC" (user control word) parameter. I am not familiar with situations where a hierarchical file organization would be useful, so I do not know how to agrue this.

Permanent files need read/write keys.
Permanent file passwords provide a minimal read or write key capability. What is missing is the ability to have one password which grants read permission and another password which grants write permission. NOS provides, instead, the ability to permit some user numbers to have read access to a file and other user numbers to have write access to that file. The file security is then provided through the security of the user numbers. This system is better than having only read/write keys because it makes it impossible for users who have not been explicitly permitted to the file in the appropriate mode to gain illegal access to the file even if they can guess the file's password.

The system should automatically search all named packs to find permanent files.
The problem is that different named packs have different characteristics: Some are shared, some are up during systems time, some are not backed up on tape, and some are not always mounted. Because of the different characteristics, it is important that the user know at all times what named pack is being used. If the system automatically searched all named packs to locate a file, the actual pack being used would be hidden from the user. Furthermore, this increased searching would seriously hurt system performance because for every GET or ATTACH request the system would have to search the default pack and all the named packs.

Files should be identified by logical type in addition to their file names.
This would increase operating system complexity while adding a capability which is easily simulated by using a simple naming convention for files. Furthermore, this suggestion runs counter to the way NOS handles files, which is to allow any file to contain any kind of data.

Multiple users within one batch account should be identified to the system individually.
The problem and solution are both administrative. The solution is to give each user his own user number. The operating system is capable of handling 131000 user numbers. So, giving each user his own user number is not impossible. The only problem is the administrative one of assigning the user numbers to the users and performing the required accounting.

There should be a way to save and/or replace all local files.
If there was a command to do this, it would also save all the scratch files left around by system routines and all the programs attached by CALLPRG. I can see no use for this capability.

XEDIT needs a move command.
It would be almost impossible to include every feature of every editor in XEDIT. The lack of a move command is probably one of the lesser deficiencies of XEDIT since a move can be easily simulated with a COPYD and a READ. There are at least two "modern" text editors which also lack a move command - the Honeywell EDIT (which XEDIT was patterned after) and the CDC's EDIT.

NOS needs transparent timesharing commands.
CDC uses transparent commands to issue commands to the 2551 front end when running under IAF/NAM. It would be confusing to add a new meaning to transparent commands.

RFL should be handled automatically.
Slowly, most of the operating system and compilers are being converted to automatically manage their own field length and thus eliminate the need for the RFL command.

NOS should automatically search for a permanent file...
If the system automatically gets a permanent file when a program attempts to access a non-existant local file, the permanent file system blurs with the local file system. That is, the user will view the permanent file system as an extension to the local file system. This makes it easier to unintentionally attach and modify permanent files. I prefer the simple straightforward way the system works now where an explicit command is required to get or attach a permanent file.

COPY, COPYCF, and COPYBF print only EOI ENCOUNTERED...
In fact, COPYCF issues the message "END OF INFORMATION ENCOUNTERED" and COPY and COPYBF issue the message "INPUT FILE NOT FOUND - name" when the input file is missing. Unfortunately, the "INPUT FILE NOT FOUND" message is usually hidden from the timesharing user by the "EOI ENCOUNTERED" message. This is probably a bug and should be fixed.

When OLD changes the subsystem a message should be issued.
The change of subsystem is a logical and expected result of the OLD command. If the system issued a warning message every time a command had a logical side effect, the user would be inundated by messages.

//////////

Cyber 74/172 Deadstart Dump Analysis from Friday, 6 June through Thursday 19 June - by K. C. Matthews

Tuesday, 10 June

20:08 (DD2004) Cyber 74
The shared queue pack, SHA, filled up. Export kept on calling PP program QAP in response to station attempts to read in a job. Eventually, the 74 was out of PP's since all copies of QAP were requesting shared queue tracks. Solutions:

1. As soon as we can, we will have another shared queue device. Having only one of any critical resource can cause problems.
2. Someone could fix Export or QAP.